

# Iterative Joint Design of Source Codes and Multiresolution Channel Codes

Andrea Goldsmith\*

Michelle Effros†

*California Institute of Technology*

## Abstract

We propose an iterative design algorithm for jointly optimizing source and channel codes. The joint design combines channel-optimized vector quantization (COVQ) for the source code with rate-compatible punctured convolutional (RCPC) coding for the channel code. Our objective is to minimize the average end-to-end distortion. For a given channel SNR and transmission rate, our joint source and channel code design achieves an optimal allocation of bits between the source and channel coders. This optimal allocation can reduce distortion by up to 6dB over suboptimal allocations for the source data set considered. We also compare the distortion of our joint iterative design with that of two suboptimal design techniques: COVQ optimized for a given channel bit-error-probability, and RCPC channel coding optimized for a given vector quantizer. We conclude by relaxing the fixed transmission rate constraint and jointly optimizing the transmission rate, source code, and channel code.

## 1 Introduction

An end-to-end communication system is composed of a system encoder, which maps the source symbols into channel inputs, and a system decoder, which maps the channel outputs into noisy reproductions of the original source symbols. The system encoder can be further broken down into a source encoder, which maps the source symbols into an intermediate alphabet, typically a set of binary strings, and a channel encoder, which maps the binary strings into coded bits or waveforms for transmission over the channel. Similarly, the system decoder can be broken down into a channel decoder and a source decoder corresponding to the respective channel and source encoders. Any system encoder-decoder pair can be represented in this manner, although the breakdown is not unique [1].

Shannon's classical separation result states that we can optimize the end-to-end system design by separately optimizing the source encoder-decoder pair and the channel encoder-decoder pair [2]. However, this result holds only in the limit of infinite source code dimension and infinite channel code block length. For practical systems, a joint source and channel code design may reduce distortion, complexity,

and delay. In this work we propose an iterative code design which jointly optimizes the source and channel codes to minimize end-to-end distortion.

Our joint code design uses COVQ for the source code and RCPC coding for the channel code. COVQ is a vector quantizer (VQ) that has been optimized for a given set of crossover probabilities of the source codeword indices [3]. The codeword indices are generally mapped to binary strings before channel encoding. Thus, the index crossover probabilities depend on the error probabilities of bits in the binary strings. However, bit errors in different locations of a binary string cause different amounts of distortion. Thus, it is desirable for the channel code to provide different levels of error protection for different bits, which can be done using a multiresolution channel code. The design of the multiresolution channel code should be matched to the COVQ design to minimize distortion. Multiresolution channel coding can be implemented using multiplexed convolutional or block codes, trellis codes [4], or RCPC codes [5]. We use RCPC codes for our channel code, although our design technique is easily extended to other forms of multiresolution channel coding.

The designs of the COVQ and RCPC codes are not independent. The optimal COVQ is the COVQ matched to the index crossover probabilities determined by the RCPC channel code. Likewise, the optimal RCPC code is the RCPC code that minimizes the expected distortion of the COVQ. The design algorithm proposed in this work achieves a joint optimization of these source and channel codes using an iterative descent technique reminiscent of the generalized Lloyd algorithm [6]. Our joint design also optimizes the bit allocation between the source and channel codes for a given channel SNR. For the source data set considered, a suboptimal allocation can increase distortion by up to 6dB.

We compare the distortion of our joint iterative design with that of two suboptimal design techniques: COVQ optimized for a given equal-error-protection convolutional channel code, and RCPC channel coding optimized for a VQ design based on a noiseless channel. These techniques work almost as well as our joint design when their bit allocations are optimized, but can exhibit large distortion increases for suboptimal bit allocations. Further distortion reduction can be achieved when the link-layer implementation (e.g., the modulation, channel transmission rate, etc.) is taken into account [7, 8]. We consider this effect by optimizing the channel transmission rate jointly with the source and channel codes. In our experiments, a suboptimal choice of trans-

\*Supported in part by ONR grant NAV-5X-N149510861.

†Supported in part by NSF CAREER Award MIP-9501977.

mission rate increases distortion by up to 4dB.

The remainder of this paper is organized as follows. The end-to-end system model is presented in Section 2. The COVQ, RCPC channel codes, and joint iterative code design are described in Section 3. Experimental results appear in Section 4, where we also determine the distortion reduction obtained by optimizing the channel transmission rate along with the joint code design. Section 5 summarizes our results and conclusions.

## 2 System Model

A block diagram of the end-to-end communication system is shown in Figure 1. We assume a discrete-time, real-valued, stationary source. The source encoder maps the set of possible  $k$ -dimensional source vectors  $\underline{x}$  into a set of binary strings  $\underline{v}$ . An example of a source vector would be a block of  $k$  pixels from an image. The binary strings can be fixed-length or variable-length, corresponding to a fixed-rate or variable-rate source code. Due to finite bandwidth constraints, the source encoder typically introduces some distortion. While variable-rate source codes achieve better compression for a given distortion and source vector dimension, they typically perform poorly when channel errors are introduced [9]. Thus we consider only fixed-rate vector quantizers, which produce fixed-length binary strings of length  $kR_s$  for each source vector  $\underline{x}$ . The number of bits per source symbol,  $R_s$ , is a parameter of the source code design. The channel encoder operates on the vector  $\underline{v}$  of  $kR_s$  bits to obtain  $k(R_s + R_c)$  coded bits  $\underline{w}$ . These coded bits (channel symbols) are then modulated and transmitted over an additive white Gaussian noise (AWGN) channel at a rate of one channel symbol per  $T$  seconds, where  $T$  is the channel symbol time.

At the receiver, the signal is first demodulated, which yields the noisy channel symbols  $\hat{\underline{w}}$ . These channel symbols are passed through the channel decoder, which may correct some (or all) of the channel errors to obtain a noisy reproduction  $\hat{\underline{v}}$  of the original binary string  $\underline{v}$ . Finally,  $\hat{\underline{v}}$  is passed through the source decoder to obtain a noisy reproduction  $\hat{\underline{x}}$  of the original source vector  $\underline{x}$ .

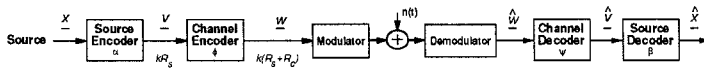


Figure 1: System Model.

The total end-to-end distortion of the system is the mean-squared-error between the source vector  $\underline{x}$  and source reproduction vector  $\hat{\underline{x}}$ :  $d(\underline{x}, \hat{\underline{x}}) = \|\underline{x} - \hat{\underline{x}}\|^2$ . For a given  $R_s$ , the total distortion decreases as  $R_c$  increases. Similarly, for a given  $R_c$ , the total distortion decreases as  $R_s$  increases. For  $R = R_s + R_c$  fixed, there is an optimal way to divide the transmission rate  $R$  between the source and channel rates  $R_s$  and  $R_c$  to minimize total distortion. This optimal allocation depends on the SNR per channel symbol  $E_s/N_0$ <sup>1</sup>

<sup>1</sup>The  $E_s/N_0 = PT/N_0$ , where  $P$  is the received signal power,  $N_0$  the spectral noise density, and  $T$  the channel symbol time.

and the source statistics. For example, channel coding is less important on high-SNR channels, so the  $R_s$  value corresponding to the optimal bit allocation will generally increase as a function of  $E_s/N_0$ . For our joint code design we consider all  $(R_s, R_c)$  pairs such that  $R_s + R_c = R$  and choose the pair and corresponding joint code with minimal distortion.

The information rate  $I_r$  of the system is defined as the number of source symbols transmitted per second. If the channel symbol rate is  $1/T$  channel symbols per second, then the information rate  $I_r = [T(R_s + R_c)]^{-1}$  source symbols per second. Thus,  $R = R_s + R_c$  is constrained by the required information rate and the channel symbol time. The information rate  $I_r$  is generally fixed. We initially assume that the modulation is also fixed, so  $T$  and therefore  $R$  are fixed parameters. We later allow  $T$  to vary, which gives one more degree of freedom in the joint source and channel code design.

## 3 Joint Iterative Code Design

The goal of our joint code design is to minimize the expected distortion  $D = E[d(\underline{x}, \hat{\underline{x}})]$  of the COVQ and RCPC codes, where the expectation is with respect to a source data training set. We use an iterative design technique to obtain this minimization. Specifically, we alternately optimize the COVQ for a given RCPC code and then the RCPC code for a given COVQ. The flow chart for our design algorithm is shown in Figure 2. This iterative approach, which has been used previously for design of joint source and channel codes [8, 10], is guaranteed to converge, since our nonnegative distortion measure is reduced at every iteration. We now describe each of the algorithm steps in Figure 2.

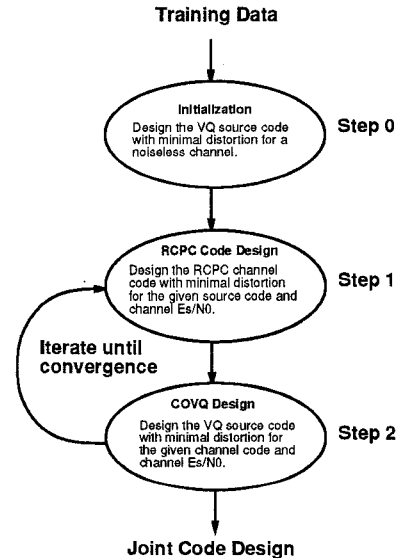


Figure 2: Iterative Code Design Process.

### 3.1 Step 0: Initialization

For a given  $R_s$  value, the training data is used to design a fixed-rate VQ with minimal distortion, assuming a noiseless

channel. The VQ is designed using the generalized Lloyd algorithm described in Section 6.4 of [6]. This yields the initial source code  $(\alpha, \beta)$  in Figure 1.

### 3.2 Step 1: RCPC Channel Code Design

In Step 1 of the algorithm, the RCPC code with minimal distortion for the VQ designed in Step 0 is found. We use the set of channel code rates from Table 1 in [5] to obtain our RCPC code candidates. Each candidate RCPC code provides a set of  $kR_s$  error protection levels for the  $kR_s$  bits in  $\underline{v}$ . We represent each candidate code by a vector of length  $kR_s$ , where the  $i$ th vector element corresponds to the channel code rate applied to the  $i$ th bit in  $\underline{v}$ . For example, for  $kR_s = 4$ , the RCPC code  $(1/2, 1/4, 1, 1)$  applies a rate  $1/2$  convolutional code to the first bit in  $\underline{v}$ , a rate  $1/4$  convolutional code to the second bit in  $\underline{v}$ , and no channel code to the third and fourth bits in  $\underline{v}$ .

The minimum-distortion RCPC code is obtained as follows. For a given VQ and corresponding  $R_s$  value, we search over all RCPC code vectors that satisfy the transmission rate constraint  $R = R_s + R_c$ . For a given  $R_s$ , an RCPC code vector  $(c_1, \dots, c_{kR_s})$  satisfies this constraint if  $\sum_{i=1}^{kR_s} (c_i)^{-1} \leq kR$ . We call an RCPC code that satisfies this constraint an eligible channel code. Note that the channel code vector  $(1/2, 1/4, 1, 1)$  satisfies this constraint for  $kR = 8$ , as would the code vectors  $(1/2, 1/2, 1/2, 1/2)$  and  $(1/3, 1, 1, 1/3)$ , along with many others. Since the VQ does not order its codeword bits relative to their error sensitivity, we must search over all eligible channel codes so that the best protection levels under the transmission rate constraint  $R$  are applied to  $\underline{v}$ . For each eligible channel code, we compute the average distortion  $D$  of the VQ with this RCPC channel code. The average distortion depends on the bit-error-rates (BERs) associated with the RCPC code, which in turn depend on the channel  $E_s/N_0$  and the modulation. We assume BPSK modulation, so the BERs associated with each RCPC code under consideration are functions of  $E_s/N_0$  and the channel code rates. Figure 5 of [5] shows a plot of BER versus  $E_s/N_0$  for each element of an RCPC channel code vector. After computing the average distortion for all eligible channel codes, we select the RCPC code with minimal distortion. This code choice corresponds to the initial channel code  $(\phi, \psi)$  in Figure 1. The design algorithm then passes to Step 2.

### 3.3 Step 2: COVQ Design

In Step 2 of the design algorithm, the VQ  $(\alpha, \beta)$  is redesigned using the RCPC channel code obtained in Step 1. This design process entails an optimization of the source encoder  $\alpha$  and decoder  $\beta$  for the codeword index crossover probabilities resulting from the RCPC channel code  $(\phi, \psi)$ . The optimal  $\alpha$  and  $\beta$  are obtained through the COVQ design algorithm described in [3]. This design technique is itself an iterative algorithm which successively redesigns  $\alpha$  for a given  $\beta$  and RCPC code, and  $\beta$  for a given  $\alpha$  and RCPC code. More details of this design process can be found in [1].

### 3.4 Design Iteration

At the conclusion of Step 2 a new source code  $(\alpha, \beta)$  has been designed for the RCPC channel code obtained in Step 1. The iterative design process then returns to Step 1 to determine the minimum-distortion RCPC code  $(\phi, \psi)$  for this new source code  $(\alpha, \beta)$ . Once a new RCPC code is obtained, this channel code is passed to Step 2 of the algorithm to obtain a new source code. Successive application of Steps 1 and 2 results in a sequence of source codes  $\{(\alpha, \beta)\}$  and corresponding channel codes  $\{(\phi, \psi)\}$  for which the average distortions form a positive nonincreasing sequence which has to converge. At convergence, the source code  $(\alpha, \beta)_{R_s}^*$  and channel code  $(\phi, \psi)_{R_s}^*$  with minimal distortion for the given  $R_s$  value are obtained. The design process is repeated for each  $R_s$  value, and the source code  $(\alpha, \beta)^*$  and channel code  $(\phi, \psi)^*$  corresponding to the  $R_s$  value with minimal distortion comprise the final joint code.

### 3.5 Suboptimal Code Designs

For comparison we consider two suboptimal code designs. The first design follows Steps 0-1 of Figure 2 and then stops, i.e., the iterative design process is eliminated. We call this a *source-optimized channel code*, since we optimize the channel code design to a source code designed for a noiseless channel.

The second technique follows the same design procedure in Figure 2 except that the channel code is restricted to an equal-error-protection channel code, i.e., the error protection levels (BERs) for all bits in  $\underline{v}$  are the same. Thus, Step 1 of Figure 2 need not search over eligible channel codes since, for each  $R_s$  value, there is only one equal-error-protection channel code with maximal error protection. We call this technique a *channel-optimized source code*, since the source code is optimized for a single channel bit-error-probability. The channel-optimized source code is a COVQ with the index crossover probabilities of the source code computed from the equal-error-protection channel code, modulation type, and channel  $E_s/N_0$ .

The joint design described in Sections 3.1-3.4 will always have less distortion than either of these suboptimal techniques, since the channel-optimized source code design and the source-optimized channel code design are subsets of the joint design process.

## 4 Experimental Results

The joint code design process along with the suboptimal design processes described in Section 3.5 were implemented in C++ and run for a range of channel  $E_s/N_0$  values. Although the iterative design process is computationally complex (approximately five iterations are required for convergence), it is done off-line. Thus, the code design complexity does not impact the system operation. For a channel with variable  $E_s/N_0$ , code designs corresponding to several  $E_s/N_0$  values can be stored, and the joint source and channel code adapted to the channel quality.

Our experimental results are computed for a test data set of 5 magnetic resonance images (MRIs) applied to the joint

code designed from a training data set of 20 other MRIs. We define the ratio of signal power to quantization and channel noise power (SQCNr) as  $\text{SQCNr (dB)} = 10 \log(\sigma^2/D)$ , where  $D$  is the distortion of the joint code averaged over the test data set and  $\sigma^2$  is the distortion of a rate zero ( $R_s = 0$ ) VQ averaged over the test data set. In Figure 3 we show a three-dimensional plot of SQCNr for our joint code design as a function of  $R_s$  and the channel  $E_s/N_0$ . We consider  $0 \leq R_s \leq 2$  bits per pixel (bpp) and  $0 \leq E_s/N_0 \leq 4$  dB. We see from this curve that improper choice of  $R_s$  can reduce the SQCNr by more than 10dB. As expected, the  $R_s$  value that maximizes SQCNr (minimizes distortion) increases as the channel  $E_s/N_0$  increases, since fewer redundant bits are needed for channel coding. This trend is illustrated more clearly in Figure 4, where we plot both the optimal  $R_s$  and the SQCNr of the corresponding joint code as a function of  $E_s/N_0$ . An exception to this trend occurs at  $E_s/N_0 = 1$  dB, where the optimal  $R_s$  value decreases. This exception is due to the behavior of RCPC codes at low  $E_s/N_0$  values, where code rates in the RCPC code with one or two redundant bits exhibit a negative coding gain [5, Figure 5]. We see that the SQCNr increases monotonically as  $E_s/N_0$  increases, with most of the transmission rate  $R$  allocated to the source code at  $E_s/N_0 = 4$  dB.

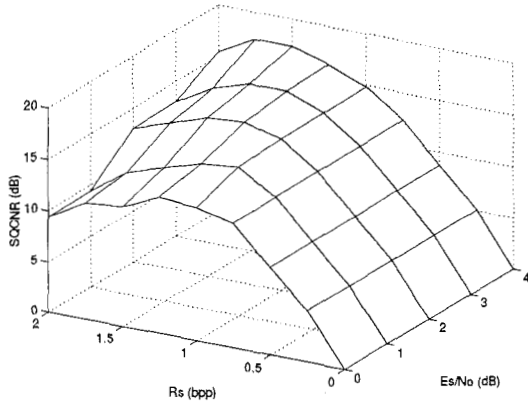


Figure 3: Joint Code SQCNr versus  $E_s/N_0$  and  $R_s$ .

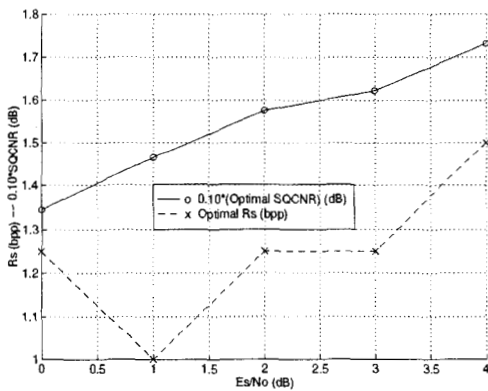


Figure 4: SQCNr and Optimal  $R_s$  versus  $E_s/N_0$ .

A comparison of SQCNr versus  $R_s$  for the joint code and the suboptimal codes described in Section 3.5 is shown in Figure 5 for  $E_s/N_0 = 2$  dB. Similar curves were obtained for  $0 \leq E_s/N_0 \leq 4$  dB. We see from this curve that the optimal bit allocation ( $R_s, R_c$ ) for the suboptimal techniques is the same as that of the joint code design. For the range of  $E_s/N_0$  values that we considered, the optimal rate allocations and the corresponding SQCNrs were approximately the same for all the joint code design techniques. Thus, the most important aspect of a joint source and channel code design is choosing the appropriate bit allocation between the source and channel codes. For each technique the SQCNr can decrease by more than 10dB with a suboptimal allocation. The optimal bit allocation depends on the channel code, the modulation, and the channel  $E_s/N_0$ . Thus, all of the design techniques described in this paper require knowledge of the link-layer implementation for good performance.

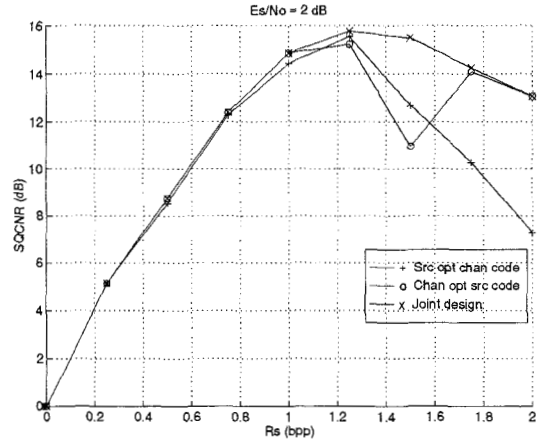


Figure 5: SQCNr versus  $R_s$  for all Design Techniques.

The distortion  $D$  at  $E_s/N_0 = 2$  dB for each technique is shown by the solid lines in Figure 6. The dashed lines in Figure 6 show the distortion contribution of the source code alone, based on the test data set and a noiseless channel. Note that for all design techniques, most of the distortion at the optimal bit allocation ( $R_s, R_c$ ) is contributed by the source code. This explains why the minimal distortions of all the joint code designs are roughly the same.

The joint source and channel code design described in Section 3 assumes that  $E_s/N_0$  is a fixed parameter of the channel. We can vary the  $E_s/N_0 = PT/N_0$  for a fixed transmit power  $P$  by changing the symbol time  $T$  to  $\hat{T}$ . However, to maintain the same information rate  $I_r = (TR)^{-1} = (\hat{T}\hat{R})^{-1}$  source symbols per second, we must also change the channel symbol rate from  $R$  to  $\hat{R} = RT/\hat{T}$ . In Figure 7 we show the change in SQCNr obtained by varying the symbol rate  $\hat{T}$  over a range of values or, equivalently, varying the effective  $E_s/N_0 = P\hat{T}/N_0$ . The calculations were done for  $E_s/N_0 = 1$  dB. Reducing the symbol time ( $\hat{T} < T$ ) causes a corresponding increase in  $R$ . This typically increases the SQCNr, since the extra bits available for the source and channel codes more than compensate for the decrease in ef-

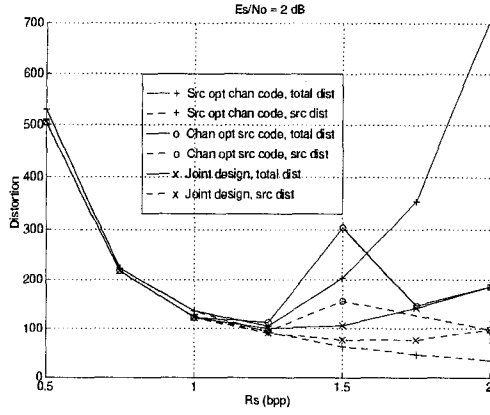


Figure 6: Distortion versus  $R_s$  for all Design Techniques.

fective  $E_s/N_0$ . However, if  $\hat{T}$  is reduced too far below its optimal value then the SQCNr decreases dramatically. Increasing the symbol duration well above the optimal value also dramatically decreases SQCNr. Note that the SQCNr is not a smooth function. We attribute this to the discrete nature of the eligible channel codes, so that the number of eligible codes changes at discrete values of  $\hat{T}$ , whereas the effective  $E_s/N_0$  is a continuous function of  $\hat{T}$ .

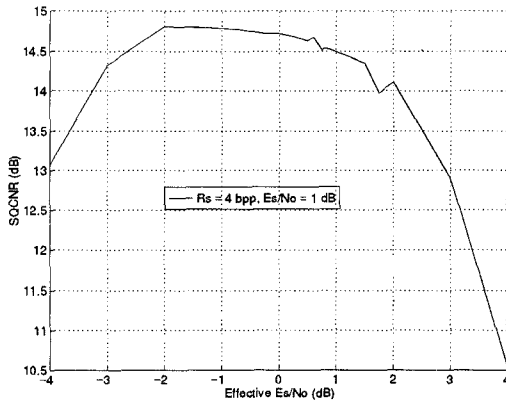


Figure 7: SQCNr versus Effective  $E_s/N_0$ .

## 5 Summary

We have described an iterative procedure for jointly optimizing a VQ source code and an RCPC channel code. The procedure optimizes the bit allocation and the joint code design to minimize end-to-end distortion. Although the computational complexity of the code design is quite high, it is done off-line, and does not impact system operation. Our joint code design outperforms source-optimized channel codes and channel-optimized source codes. The most important design aspect for any of these source and channel code designs is the optimal bit allocation between the source and channel coders. This optimal allocation depends on the source statistics, the channel quality, and the link-layer implementation. Our experiments indicate that suboptimal choice of this bit

allocation decreases SQCNr by more than 6dB.

Our distortion results indicate that multiresolution channel coding does not significantly reduce distortion relative to equal-error-protection channel coding when the bit allocation is optimized. We attribute this to the fact that the bits in the encoded source vector have approximately the same sensitivity to channel errors. We are now investigating the use of two-stage vector quantization in our design algorithm [11]. Since the codebook index (1st stage), which is very sensitive to channel errors, is sent along with the code-words (2nd stage), we expect multiresolution channel coding to have more of an impact for this source code. We are also considering the use of variable-rate modulation in our joint code design[12].

## References

- [1] A. Goldsmith and M. Effros, "Iterative joint design of multiresolution source and channel codes." *In preparation*.
- [2] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," *IRE Natl. Conv. Rec, Part 4*, pp. 142–163, 1959.
- [3] N. Farvardin and V. Vaishampayan, "On the performance and complexity of channel-optimized vector quantizers," *IEEE Trans. Inform. Theory*, Vol. 37, pp. 155–160, Jan. 1991.
- [4] A.R. Calderbank and N. Seshadri, "Multilevel codes for unequal error protection," *IEEE Trans. Inform. Theory*, Vol IT-39, No. 4, pp. 1234–1248, July 1993.
- [5] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Commun.*, Vol. 36, No. 4, pp. 389–400, April 1988.
- [6] A. Gersho and R.M. Gray, *Vector Quantization and Signal Compression*. New York: Kluwer Academic. 1992.
- [7] N. Farvardin and V. Vaishampayan, "Optimal quantizer design for noisy channels: an approach to combined source-channel coding," *IEEE Trans. Inform. Theory*, Vol. 33, pp. 827–838, Nov. 1987.
- [8] E. Ayanoglu and R.M. Gray, "The design of joint source and channel trellis waveform coders," *IEEE Trans. Inform. Theory*, Vol. 33, No. 6, pp. 855–865, Nov. 1987.
- [9] W.M. Lam and A.R. Reibman, "An error concealment algorithm for images subject to channel errors," *IEEE Trans. Image Proc.*, Vol. IP-4, No. 5, pp.533–542, May 1995.
- [10] V.A. Vaishampayan and N. Farvardin, "Joint design of block source codes and modulation signal sets," *IEEE Trans. Info. Theory*, Vol. IT-38, No. 4, pp. 1230–1248, July 1992.
- [11] P.A. Chou, M. Effros, and R.M. Gray, "A vector quantization approach to universal noiseless coding and quantization," *IEEE Trans. Inform. Theory*, Vol. IT-42, No. 4, pp. 1109–1138, July 1996.
- [12] A. Goldsmith. "Joint source/channel coding for wireless channels," *IEEE Vehic. Technol. Conf. Rec.*, pp. 614–618, July 1995.